Poster paper in the 20th Annual NACCQ, Mann, S. & Bridgeman, N. (eds) Nelson NZ, July 8-11 2007 www.naccq.ac.nz

Assessing the Steps on the Road to Relational Thinking

Anne Philpott, Phil Robbins[†] and Jacqueline L. Whalley
School of Computing and Mathematical Sciences
Auckland University of Technology

Introduction

Little is known about programming students' progression from novice to expert. Studies of the differences between novices and experts indicate that expert programmers form representations that are abstract (i.e. a relational view, Table 1) whereas novices form a concrete view (i.e. a multistructural/ unistructural view, Table 1). Lister et al. (2006) proposed that relational level reasoning in students should be encouraged because it is a more expert way of approaching code comprehension tasks. In this work we attempt to discover how we can enable and assess each students' ability to reach a relational level of reasoning. In the process we hope to identify some key steps along the road to relational thinking. We propose that code tracing is a traffic light on this road. The light is *green* if the students have progressed to relational thinking. *Orange* lights indicate students that at times display relational thinking. *Red* lights or the apparent failure of the power supply are warnings that should stop the students until the light turns *orange.*

Table 1: SOLO classifications (adapted from Biggs & Collis (1982))

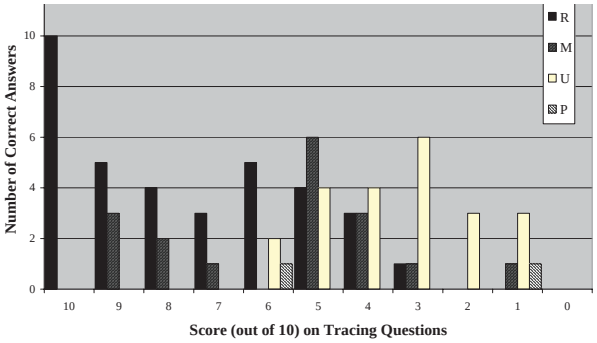
SOLO category	Description
Relational [R]	A precise summary of what the code does as a whole; it focuses on the overall purpose of the code
Multistructural[M]	A line-by-line description of the code that focuses on more than one of the language constructs.
Unistructural [U]	A description that focuses on one language construct
Prestructural [P]	A description that has no direct relationship to the code.

Analysis & Results

Eighty six examination scripts were analysed from a first semester (S1) programming paper. We examined student responses from 10 tracing questions and 2 relational questions and classified the relational questions as shown in Table 1. Unlike previous studies (Lister et al, 2006), students were primed as to how to respond to a relational question and were aware that more marks were awarded for a relational response. Table 2 displays a comparison of success in tracing with the type of SOLO response given for the relational questions.

[†]phil.robbins@aut.ac.nz

Table 2: Answer type as a function of Tracing Score



The prevalence of relational answers for those with high tracing scores clearly establishes a link between well developed tracing skills and the ability to think relationally. Conversely students who were unable to arrive at a correct answer when tracing code, displayed unistructural thinking in their relational question responses.

Leading Students down Dead End Streets

Lister et al (2006) argue that in terms of the SOLO taxonomy, a first semester examination should ensure that students succeed via relational responses. One question in our examination asks students to think relationally by naming a method. They are then required to predict the outcome of running the code which, in fact, has a bug. This bug was missed by 50% of the students who gave a correct method name. Having encouraged a relational answer to the first part of the question, students are then punished for their assumption that the code implements the identified purpose. Perhaps such questions should be avoided in assessment or, at the very least, the change in activity from identifying purpose to debugging should be more clearly sign-posted.

Conclusion

The *green* light for relational thinking would seem to be a complete mastery of the code tracing task. A better than 50% performance on the tracing task could be viewed as an orange light. However if tracing ability is at a lower level than 50% then unistructural thinking predominates and the light is definitely *red* when it comes to relational thinking.

References

Biggs, J. B. & Collis, K. F. (1982) *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome),* New York, Academic Press.

Lister, R., Simon, B., Thompson, E., Whalley, J. L., & Prasad, C. (2006). Not seeing the forest for the trees: novice programmers and the SOLO taxonomy, *Proceedings of the 11th annual SIGCSE conference on Innovation and Technology in Computer Science Education*. Bologna, Italy. 118-122. ACM Press